International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 5, October 2023, pp. 5483~5490
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i5.pp5483-5490

A profiling-based algorithm for exams' scheduling problem

Tawfiq Alrawashdeh, Khaldun G. Al-Moghrabi, Ali M. Al-Ghonmein
Faculty of Information Technology, Al-Hussein Bin Talal University, Ma'an, Jordan

Article Info

Article history:

Received Dec 8, 2022
Revised Apr 2, 2023
Accepted Apr 7, 2023

Keywords:

Decision support systems
Grouping-based approach
NP-hard problem
Scheduling algorithm
Timetable

ABSTRACT

Typically, the problem of scheduling exams for universities aims to determine a schedule that satisfies logistics constraints, including the number of available exam rooms and the exam delivery mode (online or paper-based). The objective of this problem varies according to the university's requirements. For example, some universities may seek to minimize operational costs, while others may work to minimize the schedule's length. Consequently, the objective imposed by the university affects the complexity of the problem. In this study, we present a grouping-based approach designed to address the problem of scheduling the exam timetable. The approach begins by profiling the courses' exams based on their requirements, grouping exams with similar requirements to be scheduled at the same time. Then, an insertion strategy is used to obtain the exam schedule while satisfying the imposed constraints of the targeted university. We applied this approach to the problem of exam scheduling at Al-Hussein Bin Talal University in Jordan and achieved a balanced exam schedule that met all the imposed constraints.

Corresponding Author:

Tawfiq Alrawashdeh
Department of Computer Science, Faculty of Information Technology, Al Hussein Bin Talal University
P.O. Box 20 Ma'an, Jordan
Email: Tawfiq@ahu.edu.jo

1. INTRODUCTION

The problem of scheduling exam timetables is a well-known NP-hard problem [1]. Generally, the objective of this problem is to minimize the cost of running the exam while satisfying logistics constraints. The cost can be considered as the total number of allocated exam rooms. Logistics constraints capture available resources (rooms and invigilators) and university requirements, such as the maximum number of exams a student can perform in a day. Such requirements and resources vary significantly between universities, which impacts the complexity of the problem. Accordingly, scheduling approaches must take into consideration the constraints imposed by the targeted university.

Several proposals have investigated the timetable scheduling problem in the literature [2]–[25]. Some of these proposals have proposed generic approaches [6], [7], [17], [25], while others have proposed solutions that address the requirements of specific universities. The mechanisms employed by these proposals can be generally categorized into nature-inspired solutions [7]–[10] and heuristic-based solutions [1], [8], [13], [16]. Nature-inspired solutions cover several types of techniques, such as genetic algorithms. While nature-inspired solutions have shown advantages in addressing the problem in its generic settings, heuristic solutions' capabilities to address the problem regardless of the input size have emphasized the use of heuristic approaches as a suitable technique.

In this study, we address the exam scheduling problem at Al-Hussein Bin Talal University, where paper-based and computer-based exams are typically conducted. The number of exam rooms for computer-based exams is limited compared to paper-based exams. Having two types of exams suggests that a grouping-based approach should be adopted since the requirements of these two types are different. Additionally, in this case study, some courses with different names might use the same material. These courses (synchronized courses) must be handled as one course; therefore, a single exam must be scheduled for each group of synchronized courses. Furthermore, in the presented scenario, several exams can be scheduled in the same room. These distinguishing characteristics emphasize the importance of adopting a profiling strategy, whereby each group of the courses shares the same characteristic (profile) and is scheduled separately.

Accordingly, in this study, we present a grouping-based algorithmic solution to address the problem. The solution consists of three steps: grouping, exam scheduling, and room scheduling. These steps jointly work to determine each day's exam schedule and the room allocated to each exam. The grouping step aims to divide the exams into groups, where each group consists of exams that must be scheduled during the same time slot. In the exam scheduling step, an insertion strategy is employed to establish a tentative schedule, whereas in the room scheduling step, the established schedule is confirmed by determining the room for each exam. To evaluate the performance of the proposed approach, we conducted several sets of experiments, and the results showed that the approach can obtain a conflict-free schedule that satisfies the imposed constraints. Moreover, we performed experiments to determine the maximum number of exams that can be scheduled during the available exam period.

The rest of the paper is organized as follows. Section 2 presents and discusses the most related proposals from the literature. In section 3, we discuss the proposed grouping-based approach. Section 4 presents the settings. In section 5, we discuss the results of the experiments. In section 6, we provide the discussion and remarks. We conclude the study in section 7.

2. RELATED WORK

Several proposals have addressed variations of the exam scheduling problem. Abdullah *et al.* [15] addressed the scheduling problem using a Tabu search-based mechanism. By utilizing the benefits of Tabu search, the mechanism improved the quality of the obtained schedule. However, increasing the number of exams is expected to challenge the efficiency of the proposed mechanism due to the computational requirements of Tabu search. Other searching-based solutions have investigated the exam scheduling problem [5]–[7], [10], [11], [25], including the use of simulated annealing [5], [11], genetic algorithms [10], [25], and Tabu search [7]. Dener and Calp [6] investigated a version of the problem where people who know each other are not expected to be in the same exam room. They proposed a genetic-based algorithm that considers students' contact information to determine any relationships between them.

Abela and Abramson [25] presented a two-stage scheduling approach to address the exam scheduling problem at the Arab East College for High Education in Saudi Arabia. The objective of the proposed approach is to minimize the number of exam conflicts and improve the schedule's quality. The first stage of the proposed approach acts as an input preparation step, while the second stage obtains the actual schedules. Yaldir and Baysal [8] proposed an evolutionary-based method consisting of two steps: data collection in the first step and obtaining the schedule using an evolution-based strategy in the second step.

Mandal and Kahar [9] proposed a two-stage heuristic approach to address the exam scheduling problem. The two stages work recursively to determine the schedule for all exams. In the first stage, a graph-based approach orders and schedules the subset of the exams, and the obtained schedule is optimized using the deluge algorithm. To examine the efficiency of the proposed approach, the authors used the Toronto benchmark datasets, and the results showed significant improvement compared to the traditional deluge approach. Similarly, in [10], the proposed solution starts by constructing an initial schedule using a greedy-based heuristic. Then, the quality of the obtained schedule is improved by using a variable bilinearization and decomposition technique.

Ayob *et al.* [12] proposed a scheduling approach to address the exam scheduling problem at Universiti Kebangsaan in Malaysia. In the proposed approach, the authors presented scheduling constraints as hard and soft constraints. The approach consists of two phases, each dealing with a different type of constraint. In the first phase, an initial schedule that satisfies the hard constraints is generated. The initial schedule is then used by the second phase, which obtains a schedule that also satisfies the soft constraints. The use of hard and soft constraints was also investigated by [13], whereby Abayomi-Alli [13] addressed the exam scheduling problem at the Federal University of Agriculture in Nigeria. To address this problem, the authors proposed a particular swarm optimization-based algorithm, which uses a local searching technique to improve the performance of the particle swarm method.

As discussed in this section, the scheduling problem's complexity is impacted by scheduling constraints that capture universities' resources and requirements. Some proposals have focused on reducing the violations of soft constraints based on available resources, assuming that exam rooms and invigilators are not enough to satisfy both the hard and soft constraints within a predetermined schedule length. Other proposals have investigated scheduling exams without any assumptions about resource availability. In this work, we address the exam scheduling problem at Al-Hussein bin Talal University in Jordan, which is distinguished by exam types, required schedule length, and resource availability.

3. PROBLEM DEFINITION

The exam scheduling problem can be represented as a directed acyclic graph (DAG), $G = (V, E)$. Set $V = \{C \cup S\}$ represents the courses *(C)* and the students (S) registered in these courses. Each course is associated with two variables that represent the number of students registered in this course (*sn*) and the exam delivery nature (*en*); paper-based (p) or computerized (o). Set E captures the courses registered by each student, where for each direct edge $e(v_i', v_j)$, vertex $v_i' \in S$ and set $v_j \in C$. Additionally, we are given set $T = \{t_1, t_2, t_3, \dots, t_n\}$ that represents the available exam time slots. Each time slot ($t_i \in T$) is associated with an exam room number, day (d_i). Each time slot has a period $[ST: ET]$ that represents the start and end time of the time slot. Each exam room has a capacity (*M*) that refers to the maximum number of students who can use the exam room at any given time. An exam room could be either a computer laboratory or a standard exam room without any computerized equipment. In presented problem the objective is to determine the courses exams schedule such the operational cost (the number of used rooms) is minimized. In addition to the capacity constraints, the obtained schedule must satisfy the following constraints: i) the exam for each source must be assigned to a single time slot, ii) for any given day, the number of assigned exams to a single student must be at most two, and iii) a computerized exam should be conducted in computer laboratories.

4. ALGORITHMIC SOLUTION

The process of scheduling exams must take into consideration several challenging factors, such as the presence of synchronized courses and the delivery mode of exams (online/offline). Synchronized courses should be scheduled during the same time slot, and the delivery mode of exams has a significant impact on the exam schedule, since online exams can be scheduled in a small subset of the exam rooms. In this work, we present an approach based on profiling to determine the exam schedule. The algorithm presented consists of three main steps: the grouping step, the exam scheduling step, and the exams venue scheduling step.

4.1. Grouping step

The objective of this step is to divide the students into a set of groups, with the aim of treating students registered in a course and all of its synchronized courses as a single group. Figure 1 illustrates the process of the grouping step. It begins with retrieving the most recent course catalog, which displays the updated course names and numbers. Then, in a sequential fashion, starting from the first course, a group is constructed for that course. This group includes all students registered in the course and all of its synchronized courses. For any given course $c_i \in C$, to identify the list of synchronized courses, we begin by determining the list of all available sections for that course. Then, for each section, we identify the instructor identification number (IID), section time (SI), and location (SL) (lines). This information is used to determine all synchronized courses for course $c\_i \in C$, as any other courses given by the same instructor during the same section time will be considered synchronized courses for course $c_i \in C$, since any other courses that are given by the same instructor during the same section time will be considered a synchronized course for $c_i \in C$. The result of this step is a set of groups $G = \{g_1, g_2, g_3, \dots\}$. Therefore, a student $s_i \in S$ is a member in M number of groups, where M is the total number of courses registered by the student in the current active semester.

4.2. Exam scheduling step

Figure 2 illustrates the exam scheduling step process. At most three exam sessions are expected to be scheduled on each exam day. For each session, the algorithm tracks the number of available seats (*ns*) in the exam rooms that meet the delivery criteria (paper-based or computer-based). The largest group ($g_i \in G$) will be selected and inserted into the schedule on the first available exam session on the first exam day, provided that the available number of seats in this session is greater than or equal to the size of the group. After inserting the group's exam into the schedule, the algorithm checks whether the current state of the schedule violates the constraint of assigning more than two exams to a student on any given day. If this constraint is violated, the group's exam is rescheduled on the next available exam day. If there is no available session on the current exam day with a sufficient number of available seats, the group under consideration will be scheduled on the next available exam day. Once the exam schedule for a group ($g_i \in G$) is confirmed, the algorithm proceeds to schedule the next group while considering the first available day with a sufficient number of seats as the first possibility for the exam.

```
Grouping()
Output: G = {g1, g2, g3, ...}
    CT ← getCatalog()
    i ← 0
    while CT not empty do
        c ← getCourse(CT)
        remove c from CT
        S' ← getRegisteredStudents(c)
        add S' to gi
        C' ← getSynchronizedCourses(c, C)
        add getRegisteredStudents(C') to S'
        add S' to gi
0
        i ← i + 1
1
    return G
2
```

Figure 1. Grouping

```
examScheduling(G)
Input: G (students groups)
output: SE (exams sessions)
    day ← 1
    G ← sort(G)
    for each gi ∈ G do
        se ← findSession(gi, 1, day)
        if se is null or examsNumber(gi, day, se)
                                   > 2 then
            day ← day + 1
            createSessions(day)
            se ← findSession(gi, 1, day)
        add gi to se
        updateNS(se, gi)
0
    return SE
1
```

Figure 2. Exam scheduling

4.3. Scheduling rooms step

The objective of this step is to minimize the number of exam rooms allocated to each group $g_i \in G$. To achieve this objective, several factors, including the number of available rooms and their capacity, must be taken into consideration during the allocation process. Figure 3 illustrates the exam room allocation step. This step works iteratively to determine the exam rooms for each exam session. Starting from the first-day exam session, the allocation process orders the groups allocated to this session in descending order based on their sizes. In addition to the student groups, the available exam rooms are also ordered in a descending style based on their capacity. Then, in a sequential fashion, the student groups are assigned to the exam rooms, with the criteria of this assignment being to allocate the exam room (*s*) with the highest capacity to the larges.

```
roomsScheduling(SE, R)
Input: SE (exams sessions)
                 R (exams rooms)
output: M < G, R > rooms assignment
    SE ← orderSessions(SE)
    R ← orderRooms(R)
    for each sei ∈ SE do
        G' ← getGroups(sei)
        G' ← orderGroup(G')
        for each gi ∈ G' do
            M ← assignRoom(gi, R)
    return M
```

Figure 3. Rooms scheduling

5. EVALUATION

To evaluate the performance of the presented approach, we conducted an extensive set of experiments using input data representing the students' registration status during the first semester of the academic year (2021-2022) at Al-Hussein Bin Talal University. The summary of the input data used in the experiments is shown in Table 1. Unless mentioned otherwise, we assume that there are four available exam sessions on each exam day and a total of 50 available exam rooms, of which 10% are prepared for computerized exams and the rest are for paper-based exams. Furthermore, we assume that 15% of exams are for parallel courses, and the percentage of students with more than two exams on the same day is zero in all experiments. We also assume that the total exam period is 14 days in all experiments.

To examine the number of exams allocated per session, we conducted experiments, and the results are shown in Figure 4. Based on Figure 4, the presented algorithm exhibits a greedy behavior, where the number of exams allocated in the morning session is expected to be higher than the number of exams allocated in the evening session. This behavior is desirable when more invigilators are available during morning sessions.

Table 1. Summarize input

Number of students	computerized exams:	6220
	paper-based exams	8730
	the largest class	317
	the smallest class	12
	Average	35
Number of seats	largest exam room	287
	smallest exam room	60
	Average	82

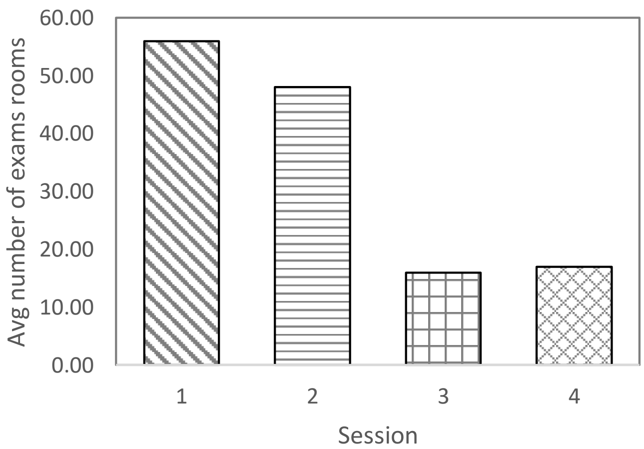

Figure 4. Exam per session

Next, we investigated the impact of students on the number of used exam rooms per session. Table 2 shows the number of students scheduled to have an exam in each session, as well as the average number of exam rooms used per session. Based on Table 2, we can see that increasing the number of students does not have a direct linear impact on the number of used exam rooms. The student's constraint of having only two exams per day, as well as the total number of students per exam, play an important role in determining the number of exams per day, which explains the observed performance.

Table 2. Exam rooms per session

Day	Number of students	Average number of sessions per room
1	2721	3
2	4788	4
3	4769	4
4	3902	4
5	1876	1
6	3317	3
7	1879	1
8	1879	1
9	1982	1
10	3512	1
11	4161	4
12	2197	2
13	2648	3
14	1654	2
15	1815	3

To study the impact of the number of students on the percentage of students with two exams on the same day, we conducted experiments, and the results are presented in Table 3. Based on the results, we can observe that the percentage of students with two exams on the same day is significantly low throughout the entire exam period. Moreover, we can also notice that there is no direct correlation between the number of students and the reported percentage. To understand this behavior, we considered the mechanism employed by the proposed algorithm. Initially, the expected exams are stored based on the number of students. Thus, exam groups with relatively similar numbers of students are expected to be allocated next to each other in the groups' list. Typically, exams for general courses are anticipated to have significantly higher numbers of students compared to specialty (major) course exams. This trend is also expected when comparing the first and second-year courses with the third and fourth-year courses, where exams for later years are expected to have fewer students. Combining this with the fact that each student is expected to be enrolled in major and general courses, the probability of having two exams on the same day is reduced since the groups of the course are distributed across the groups' list. These factors contribute to the observed behavior.

Table 3. Students with two exams

Day	Number of students	Percentage of students with 2 exams on the same day
1	2721	0.005145
2	4788	0.007728
3	4769	0.077584
4	3902	0.000256
5	1876	0.034648
6	3317	0.000603
7	1879	0.007451
8	1879	0.124002
9	1982	0.005045
10	3512	0.001708
11	4161	0.010334
12	2197	0.019572
13	2648	0.002266
14	1654	0.000605
15	1815	0.002755

6. DISCUSSION AND REMARKS

To analyze the obtained results, the development team has raised several concerns and remarks that we discuss below in detail: Firstly, it is important to ensure the integrity of exams for courses with overlapping material. Some courses are expected to have overlapping material, and if the percentage of such material is relatively high, it can introduce exam integrity issues. Therefore, further analysis should be performed to determine the percentage at which such courses should be treated as a single course. Additionally, as we have discussed, synchronized courses can add pressure to the scheduling process. Hence, eliminating such courses is expected to improve overall efficiency. Secondly, the upper bound of the number of registered students in each division is crucial in determining the quality of the obtained schedule in terms of the number of days and sessions required. Reducing the upper bound of the number of students in each division can lower the number of exams that need to be scheduled simultaneously, prevent overlapping schedules, and minimize the number of required exam days. This can ultimately lead to a more effective exam scheduling process. Lastly, the number of allocated exam halls should also impact the quality of the obtained schedules. Increasing the number of exam halls can ensure that exams are distributed in a timely and efficient manner, thereby reducing the overall duration of the exam period.

7. CONCLUSION

The resources and requirements of universities, including various constraints, have a significant impact on the complexity of scheduling exams. Therefore, in this study, we addressed the challenge of scheduling exams at Al-Hussein Bin Talal University, taking into account several constraints imposed by the university. To tackle this issue, we investigated the grouping approach, which can be considered a profiling mechanism. This approach aims to simplify the problem by scheduling exams belonging to the same group at the same time. The exam strategy employed in the scheduling process minimizes the operational cost while making efficient use of the university's resources. Our results demonstrated that increasing the number of students does not negatively impact the performance of the proposed approach.

REFERENCES

[1] O. H. Ibarra and C. E. Kim, "Heuristic algorithms for scheduling independent tasks on nonidentical processors," *Journal of the ACM*, vol. 24, no. 2, pp. 280–289, Apr. 1977, doi: 10.1145/322003.322011.

[2] I. Balan, "Using genetic algorithms to university timetabling problem," *The USV Annals of Economics and Public Administration*, vol. 22, no. 1 (35), pp. 160–166, 2022.

[3] C. H. Wong, S. L. Goh, and J. Likoh, "A genetic algorithm for the real-world university course timetabling problem," in *2022 IEEE 18th International Colloquium on Signal Processing and Applications (CSPA)*, May 2022, pp. 46–50, doi: 10.1109/CSPA55076.2022.9781907.

[4] A. A. Gozali and S. Fujimura, "Solving University Course timetabling problem using multi-depth genetic algorithm," *SHS Web of Conferences*, vol. 77, May 2020, doi: 10.1051/shsconf/20207701001.
[5] D. Abramson, "Constructing school timetables using simulated annealing: sequential and parallel algorithms," *Management Science*, vol. 37, no. 1, pp. 98–113, Jan. 1991, doi: 10.1287/mnsc.37.1.98.
[6] M. Dener and M. H. Calp, "Solving the exam scheduling problems in central exams with genetic algorithms," *Mugla Journal of Science and Technology*, vol. 4, no. 1, pp. 102–115, Jun. 2018, doi: 10.22531/muglajsci.423185.
[7] A. Hertz, "Tabu search for large scale timetabling problems," *European Journal of Operational Research*, vol. 54, no. 1, pp. 39–47, Sep. 1991, doi: 10.1016/0377-2217(91)90321-L.
[8] A. Yaldir and C. Baysal, "Developing examination scheduling automation system by using evolutionary computing technique," *Pamukkale University Journal of Engineering Sciences*, vol. 18, no. 2, pp. 105–122, 2012, doi: 10.5505/pajes.2012.91885.
[9] A. K. Mandal and M. N. M. Kahar, "Solving examination timetabling problem using partial exam assignment with hill climbing search," in *2015 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, Apr. 2015, pp. 84–89, doi: 10.1109/ISCAIE.2015.7298333.
[10] S. Wang, M. Bussieck, M. Guignard, A. Meeraus, and F. O'Brien, "Term-end exam scheduling at United States military academy/west point," *Journal of Scheduling*, vol. 13, no. 4, pp. 375–391, Aug. 2010, doi: 10.1007/s10951-009-0153-5.
[11] M. A. S. Elmohamed, P. Coddington, and G. Fox, "A comparison of annealing techniques for academic course scheduling," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1998, pp. 92–112.
[12] M. Ayob *et al.*, "Intelligent examination timetabling software," *Procedia-Social and Behavioral Sciences*, vol. 18, pp. 600–608, 2011, doi: 10.1016/j.sbspro.2011.05.087.
[13] O. Abayomi-Alli, A. Abayomi-Alli, S. Misra, R. Damasevicius, and R. Maskeliunas, "Automatic examination timetable scheduling using particle swarm optimization and local search algorithm," in *Data, Engineering and Applications*, Singapore: Springer Singapore, 2019, pp. 119–130.
[14] T. S. Nguyen, H. T. Duong, and K. D. Nguyen, "Automatic generation of course schedules using genetic algorithm," in *Artificial Intelligence in Data and Big Data Processing*, Springer International Publishing, 2022, pp. 37–45.
[15] S. Abdullah, S. Ahmadi, E. K. Burke, M. Dror, and B. McCollum, "A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem," *Journal of the Operational Research Society*, vol. 58, no. 11, pp. 1494–1502, Nov. 2007, doi: 10.1057/palgrave.jors.2602258.
[16] B. Bilgin, E. Özcan, and E. E. Korkmaz, "An experimental study on hyper-heuristics and exam timetabling," in *Practice and Theory of Automated Timetabling VI*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 394–412.
[17] E. K. Burke, D. Elliman, and R. Weare, "A genetic algorithm based university timetabling system," in *Proceedings of the 2nd east-west international conference on computer technologies in education*, 1994, pp. 35–40.
[18] C. W. Fong, H. Asmuni, P. H. Leong, Y. H. Sam, Y. Y. Pang, and H. M. Sim, "Zombie survival optimization in solving university examination timetabling problem," in *2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Jun. 2022, pp. 169–173, doi: 10.1109/I2CACIS54679.2022.9815494.
[19] L. Koppenhol, N. Brouwer, D. Dijkzeul, I. Pijning, J. Sleegers, and D. van den Berg, "Exactly characterizable parameter settings in a crossoverless evolutionary algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Jul. 2022, pp. 1640–1649, doi: 10.1145/3520304.3533968.
[20] A. Ghaffar, M. Sattar, M. Munir, and Z. Qureshi, "Multi-objective fuzzy-based adaptive memetic algorithm with hyper-heuristics to solve university course timetabling problem," *ICST Transactions on Scalable Information Systems*, Jul. 2018, doi: 10.4108/eai.16-12-2021.172435.
[21] I. A. Abduljabbar and S. M. Abdullah, "An evolutionary algorithm for solving academic courses timetable scheduling problem," *Baghdad Science Journal*, vol. 19, no. 2, Apr. 2022, doi: 10.21123/bsj.2022.19.2.0399.
[22] T. Feutrier, M.-É. Kessaci, and N. Veerapen, "Analysis of search landscape samplers for solver performance prediction on a university timetabling problem," in *Lecture Notes in Computer Science*, *Springer International Publishing*, 2022, pp. 548–561.
[23] S. T. Ngo, J. B. Jaafar, I. A. Aziz, G. H. Nguyen, and A. N. Bui, "Genetic algorithm for solving multi-objective optimization in examination timetabling problem," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 11, Jun. 2021, doi: 10.3991/ijet.v16i11.21017.
[24] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Systems with Applications*, vol. 165, Mar. 2021, doi: 10.1016/j.eswa.2020.113943.
[25] J. Abela and D. Abramson, "A parallel genetic algorithm for solving the school timetabling problem," *Division of Information Technology, CSIRO (Citeseer, 1991)*, 1991.

BIOGRAPHIES OF AUTHORS

Tawfiq Alrawashdeh serves as an assistant professor at the Department of CS, Al-Hussein Bin Talal University, Jordan. He got a bachelor's in computer science from Al Hussein Bin Talal University (AHU) Jordan in 2005. He then gets a master's in computer science from Al-Balqa' Applied Jordan 2011. He received his Ph.D. in 2020 from Universiti Sultan Zainal Abidin, Malaysia. His research interests focus on cloud computing and task scheduling. Tawfiq worked as a Head of programming division at Al Hussein Bin Talal University from 2005 to 2014. Then worked for Al Hussein Bin Talal University from 2014 to present as Lecturer where he teaches a different course in computer science collage. in additional, he starts working as Director of Computer and Information Technology Center from 2018 to present. He can be contacted at email: tawfiq@ahu.edu.jo.

Khaldun G. Al-Moghrabi serves as an assistant professor at the Department of CIS, Al-Hussein Bin Talal University, Jordan. He received his bachelor's degree in CIS from Al-Hussein Bin Talal University (2006) and holds a master's degree in CIS from the Middle East University in Jordan, 2009. In 2018, he earned his doctorate degree in MIS from OIU, Sudan. His research interests include E-learning, decision support systems, database systems, CC, big data, and the IoT. He can be contacted at email: khaldun.g.moghrabi@ahu.edu.jo.

Ali M. Al-Ghonmein serves as an assistant professor at the Department of Computer Information Systems (CIS), Al-Hussein Bin Talal University, Jordan. He received his bachelor's degree in Computer Science from Al-Hussein Bin Talal University (2004) and obtained his master's degree in CIS from the Arab Academy for Banking and Financial Sciences in Jordan, 2008. In 2018, he earned his doctorate degree in Management Information Systems (MIS) from Omdurman Islamic University (OIU), Sudan. His research interests include information retrieval, decision support systems, natural language processing, database systems, cloud computing, big data, and the IoT. He can be contacted at email: ali.m.alghonmein@ahu.edu.jo.